Moore 狀態機之 verilog 程式

一、設計分成兩部分

1.狀態變遷

2.輸出

狀態輸出(令狀態輸出為 state,輸入為 in)

```
always @ (posedge clk or negedeg rst)
if(~rst_) state=S0;//IDLE 狀態
else begin{
        if(in==state1 條件) state=S1;
        else if (in==state2  條件) state=S2;
end
```

輸出(令輸出變數為 out)

```
always @ (state ot rst_)
if(~rst_) out=0;
else begin
        case(state)
        S0:out=S0 之輸出;
        S1:out=S1 之輸出;
end
```

狀態參數

以 parameter 宣告狀態碼

例:4 個狀態以 2 位元 parameter 來宣告

```
parameter [1:0] S0=2’d0,S1=2’d1,S2=2’d2,S2=2’d3;
```

例:8 個狀態需以 3 位元 parameter 宣告

```
parameter[2:0] S0=3’d0 , S1=3’d1.....S5=3’d5,s6=3’d6,s7=3’d7;
```

以下為狀態機之 verilog 程式碼

```
module stateMachine1(clk,rst_,in,out);
input clk,rst_;
input in;
output out;
parameter [1:0] S0=2’d0,S1=2’d1,d2=s’d2,d3=2’d3;

//狀態宣告
reg[1:0] state;
always@(posedge clk or negedeg rst_)
if(~rst) state=S0;
else begin
```

```
        case(state)
        S0: if(in==1'b1) state =s2;
        S2: if(in==1'b1) state =s3;
        S3: if(in==1) state =s1;
        S1: if(in==1) state =s2;
        endcas
end
//狀態輸出
always@(state or rst_)
if(~rst) out=1'b0;
else begin
        case(state)
        S0: out=1'b0;
        S1: out=1'b0;
        S2: out=1'b0;
        S3: out=1'b0;
        Endcase
end
endmudule
```

測試程式 test.v;

```
'timescale 1ns/100ps
'include "moore1.v"

module test;
reg clk,rst_;
reg in;
wire out;

moore1 u1(clk,rst_,in,out);

always #20;
        clk=~clk;

initial begin
#0 clk=1'b0;
#1100 $finish;
end

initial begin
```

```
#0 rst_=1’b0;
#55 rst_= 1’b1;
end

initial begin
#0 in=1’b0; //S0
#60 in=1’b1; //S2
#40 in=1’b0;
end

initial begin
#0 in=1’b0; //S0
#60 in=1’b1; //S2
#40 in=1’b0;
#80 in=1’b1; //S3
#40 in=1’b0;
#80 in=1’b1; //S1
#40 in=1’b0; //S0
#80 in=1’b1; //S2
#40 in=1’b0;
#80 in=1’b1; //S3
#40 in=1’b0;
#80 in=1’b1; //S1
#40 in=1’b1; //S2
end

initial begin
$dumpfilm(“moore1.vcd”);
$dumpvars();
end

endmodule
```

狀態機設計十字路口紅綠燈(Traffic Light)

狀態 1:

poss1　R1 亮、Y1 滅、G1 滅

R2 滅、Y2 滅、G2 亮

Road 2 通行、Rood1 停止

經過 20s

狀態 2:

flash2　Road 2 閃黃燈 5s

Road 1 停止

閃黃燈 0.5s 亮、0.5s 滅

R1 滅、Y1 滅、G1 滅

R2 滅、Y2 閃爍、G2 滅

經過 5s

狀態 3:

pass1　Road 1 通行

Road 2 停止

R1 滅、Y1 滅、G1 亮

R2 亮、Y2 滅、G2 滅

經過 20s

狀態 4:

flash1　Road 1 閃黃燈 5s(如狀態 2)

經過 5s

回到狀態 1

狀態設定

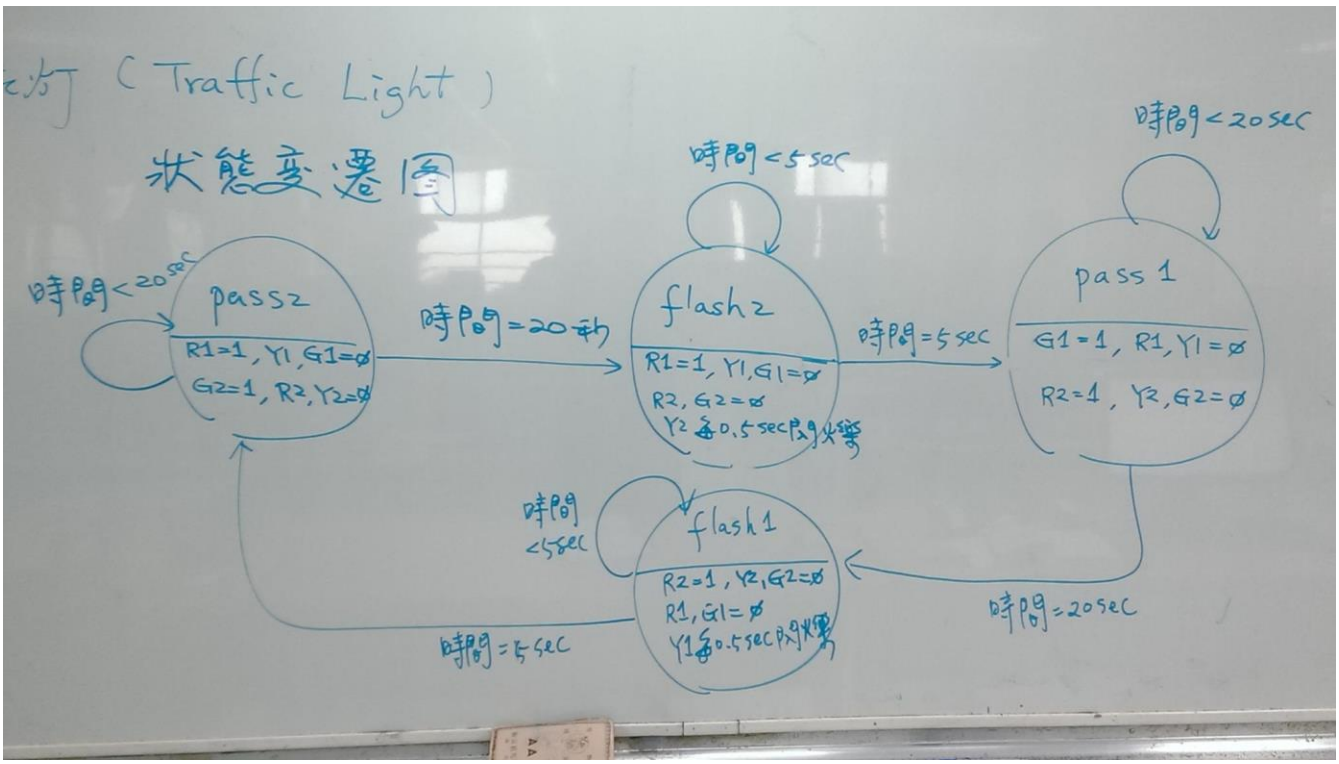
狀態 1=pass2

狀態 2=flash2

狀態 3=pass1

狀態 4=flash1

狀態變遷圖



狀態編碼

pass2=0

flash2=1

pass1=2

flash=3

每 0.5s 產生一脈衝

輸出 Road1 紅燈為 R1、綠燈為 G1、黃燈為 Y1

Road2 紅燈為 R2、綠燈為 G2、黃燈為 Y2

若輸入時脈頻率=10MHZ

0.5s=0.5/(1/10M)=5M 次時脈

需要 23 位元計數器來產生 0.5s 之脈衝

```
reg[22:0] halfSecCount;
wire halfSecCount
assign #1 halfSecPulse
          = halfSecCount=-23'd4999999);
always @ (posedge clk or negedge rst_)
if(~rst_) halfSecCount=23'd0;
else if(halfSecPulse) halfSecCount=23'd0;
else halfSecCount = #1 halfSecCount+1'd1;

parameter [1:0]
     pass2=2'd0 , flash2=2'd1 , pass1=2d'2 , flash1=2'd3;
```

```
reg[5:0] lightCount;
//lightCount 為狀態機之計時變數 , lightcount 計算 0.5s 之脈衝 , lightCount=40 表示 20s ,
lighCount=10 表示 5s

狀態變遷
reg [1:0] state;
always @(posedge clk or negedge rst_)
if (~rst_)state=pass2;
else begin
        case (state)
        pass2: if(lightCount==40)state=flash2;
        flash2:if(lightCount==10)state=pass1;
        pass1: if(lightCount==40)state=flash1;
        flash1:if(lightCount==10)state=pass2;
        endcase
end.

always@(posedge clk ot negedge rst_)
if(~rst) lightCount=6'd0;
else if ((lightCount==6'd40 && (state==pass1 || state=pass2)) || (lightCount==6'd1
            &&(state=flash1 || stat == flash2)))
            lightCount=6'd0;
else if (halfSecPulse)lightCount=#1 lightCount+1'b1;

程式:
module trafficLight(clk,rst_,R1,Y1,G1,R2,Y2,G2);
input clk,rst_;
output R1,Y1,G1,R2,Y2,G2;

reg[22:0]halfSecCount;
wire halfSecPulse;
assign #1 halfSecPulse=( halfSecCount=23'd4999999);

always @(posedge clk or negedge rst_)
if(~rst) halfSecCount=23'd0;
else if(halfSecPulse)1'd1;
else halfSecCount= halfSecCount+1'b1;

reg[5:0]lighgCount;
reg[1:0]state;
parameter [1:0] pass2=2'd0,flasg2=2d'1,pass1=2'd2,flash1=2'd3;
```

```
always @(posedge clk or negedge rst_)
if(~rst) state =pass2;
else begin
        case(state)
        pass2:if(cightCount==6'd40) state flash2;
        flash2:if(cightCount==6'd10) state pass1;
        pass1:if(cightCount==6'd40) state flash1;
        flash 1:if(cightCount==6'd10) state pass2;
        endcase
end

always@(posedge clk or negedge rst_)
if(~rst) lightCount=6'd0;
else if ((lightCount==6'd40 && (state==pass1 || state=pass2)) || (lightCount==6'd1
            &&(state=flash1 || stat == flash2)))
            lightCount=6'd0;
else if (halfSecPulse)lightCount=#1 lightCount+1'b1;

reg R1,Y1,G1,R2,Y2,G2;
always @(state or rst_)
if(~rst) begin
        R1=1'b1;Y1=1'b0;G1=1'b0;
        R2=1'b0;Y2=1'b0;G2=1'b1;
        end
else begin
        case(state)
        pass2:begin
        R1=1'b1;Y1=1'b0;G1=1'b0;
        R2=1'b0;Y2=1'b0;G2=1'b1;
        end
        pass1:begin
        R1=1'b1;Y1=1'b0;G1=1'b0;
        R2=1'b0;Y2=1'b0;G2=1'b1;
        end
        flash2:begin
            if(halfSecPulse) Y2 = #1 ~Y2;
            R1=1'b1;Y1=1'b0;G1=1'b0;
            R2=1'b0; G2=1'b0;
        end
        flash1:begin
            if(halfSecPulse) Y1 = #1 ~Y1;
            R1=1'b0; G1=1'b0;
```

```
            R2=1’b1;Y2=1’b0;G2=1’b0;
        end
        end case
end
endmodule
```

測試程式 trafficlight.v;

```
‘timescale 1ns/100ps
‘include “trafficlight.v”

module test;
reg clk,rst_;
wire R1,Y1,G1,R2,Y2,G2;

trafficlight u1(clk,rst_,R1,Y1,G1,R2,Y2,G2);

always #5;
        clk=~clk;

initial begin
#0 clk=1’b0;
#6000000000 $finish;
end

initial begin
#0 rst_= 1’b0;
#6 rst_= 1’b1;
end

initial begin
$dumpfilm(“trafficlight.vcd”);
$dumpvars();
end
endmodule
```